

REMARKS

Favorable reconsideration of this application is respectfully requested.

The specification is amended by the present response to correct for minor grammatical and idiomatic informalities. The changes made to the specification are deemed to be self-evident from the original disclosure, and thus are not deemed to raise any issues of new matter.

Claims 2, 3, 10, 11, 17, 18, and 20 are amended by the present response to address the objection noted in paragraph 1 of the Office Action.

Claims 1-24 are pending in this application. Claims 21-24 are newly added by the present response. Claims 1-20 were rejected under 35 U.S.C. § 112, second paragraph. Claims 1, 9, 16, and 19 were rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. patent 6,182,281 to Nackman et al. (herein "Nackman") in view of U.S. patent 5,613,120 to Paylay et al. (herein "Paylay"). Claims 2, 3, 10, 11, 17, 18 and 20 were rejected under 35 U.S.C. § 103(a) as unpatentable over Nackman in view of Paylay and further in view of "Access 2000 for Windows for Dummies" by John Kaufeld, 1999 (herein "Kaufeld"). Claims 4 and 12 were rejected under 35 U.S.C. § 103(a) as unpatentable over Nackman in view of Paylay and further in view of Kaufeld and U.S. patent 6,041,179 to Bacon et al. (herein "Bacon").

Addressing first the rejection of claims 1-20 under 35 U.S.C. § 112, second paragraph, that rejection is traversed by the present response. More specifically, claims 1-20 are amended by the present response to more clearly recite the features therein and to correct any grammatical and idiomatic informalities. The claim amendments also address the specific objections noted in paragraph 3 of the Office Action. The presently submitted claim amendments are believed to address the outstanding rejection under 35 U.S.C. § 112, second paragraph.

Addressing now the rejection of Claims 1, 9, 16, and 19 under 35 U.S.C. § 103(a) as unpatentable over Nackman in view of Paylay, that rejection is traversed by the present response.

Each of the above-noted independent claims is amended by the present response to clarify features recited therein. Specifically, claim 1 clarifies that the data type definition table stores “a use flag set in a use status when the corresponding data type definition is described in the body of any of all source programs to be linked to the one object program”. The operation of the “code optimizing processor” is also clarified in claim 1, particularly to clarify that the data type definition from the source programs is deleted “when a data type definition is already stored or the use flag is not set in a use status in the data type definition table ... to optimize the source program.” The other independent claims 9, 16, and 19 are similarly amended as in independent claim 1 noted above. The above-noted features are clearly supported by the originally specification, see, for example, page 16, line 23 to page 20, line 25 and Figures 14-16B.

The primary cited reference to Nackman teaches (a) a preprocessor for executing preprocessing of source programs (see column 18, lines 44-50) and (b) a language processor for compiling the source programs (see column 18, lines 3-13). The outstanding rejection is based on the position that Paylay teaches a class definition table (at column 2, lines 15-17) and a linker that removes duplicate class definitions (at column 18, lines 50-61), and that it would have been obvious to one of ordinary skill in the art at the time of the invention to develop a system with a preprocessor and compiler as taught by Nackman and include a data type definition table and a processor for removing duplicate data definitions from the code as

taught by Paylay.¹ The noted motivation for such a combination is that removing a duplicate code reduces the size of the overall program and promotes better organization of the code.²

However, with respect to the newly recited limitations in independent claims 1, 9, 16, and 19, those claims as currently written recite that a data type definition table is arranged for one object program and is configured to store a data type definition for data or a function in the source programs and a use flag set in a use status when the corresponding data type definition is described in the body of any of all source programs to be linked to the one object program. Further, with reference to the data type definition table, a code optimizing processor scans all the preprocessed source programs to be used as a source for generating the object program, and can delete, when a data type definition is already stored or the use flag is not set in the use status in the data type definition table, the data type definition from the source programs to optimize the source programs. Thereby, the code optimizing processor can delete unnecessary data type definitions in two instances, by referring to a use flag in the data type definition table arranged for one object to which a plurality of source programs are linked.

In contrast to the features now clarified in the above-noted claims, Nackman at column 18, lines 2-3 and 44-50, and Paylay at column 12, lines 15-17, and column 28, lines 50-61, fail to disclose or suggest the above-noted data type definition table containing a use flag set in a use status when the corresponding data type definition is described in the body of any of all source programs to be linked to the one object, and whereby a code optimizing processor is configured to delete unnecessary data type definition in two instances, by referring to a use flag in a data type definition table.

Moreover, Paylay merely discloses a class definition table 414 containing a list of class definitions at column 12, lines 15-17, and a linker that removes duplicate class

¹ Office Action of December 18, 2002, page 4, lines 5-10.

² Office Action of December 18, 2002, page 4, lines 10-11.

definitions between a plurality of object files 106, 108 at column 28, lines 39-61. In contrast to Paylay, in the above-noted claims as currently written, the code optimizing processor is configured to eliminate unnecessary data type definitions in two instances in one object file.

In such ways, each of amended independent claims 1, 9, 16, and 19 positively recites limitations neither taught nor suggested by the combination of teachings of Nackman in view of Paylay, and thus those claims, and the claims dependent therefrom, are allowable over the outstanding rejection.

Addressing the further rejections based on the combination of teachings of Nackman and Paylay and further in view of Kaufeld, and further view of Bacon, those rejections are also traversed by the present response. Specifically, no teachings in Kaufeld or Bacon can overcome the above-noted deficiencies with respect to Nackman in view of Paylay, and thus those further rejections are also traversed by the present response.

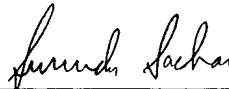
The present response also sets forth new Claims 21-24 for Examination. Support for those new claims is deemed to be self-evident from the original disclosure, including the original claims. Those new Claims 21-24 are also deemed allowable for similar reasons as discussed above.

In summary, each of the claims is amended by the present response to address the objections thereof. Each of the claims is also amended to overcome the outstanding rejection under 35 U.S.C. § 112, second paragraph. Further, each of the claims as currently written recites features neither taught nor suggested by the applied art.

As no other issues are pending in this application, it is respectfully submitted that the present application is now in condition for allowance, and it is hereby respectfully requested that this case be passed to issue.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Eckhard H. Kuesters
Registration No. 28,870
Surinder Sachar
Registration No. 34,423
Attorneys of Record



22850

Tel: (703) 413-3000
Fax: (703) 413 -2220
EHK/SNS/cja

I:\ATTY\SNS\19's\193857\193857US-AM.DOC